

**VYSOKÁ ŠKOLA BÁŇSKÁ
TECHNICKÁ UNIVERZITA OSTRAVA**

**Fakulta elektrotechniky a informatiky
Katedra informatiky**

**Evidenční systém majetku
Register system of assets**

Rok 2010

Lukáš Veselý

Zadání bakalářské práce

Student:

Lukáš Veselý

Studijní program:

B2647 Informační a komunikační technologie

Studijní obor:

2612R025 Informatika a výpočetní technika

Téma:

Evidenční systém majetku
Register System of Assets

Zásady pro vypracování:

Cíl práce je návrh a alespoň částečná realizace jednoduchého systému, který by umožnil evidovat majetek v menších organizacích.

1. Systém by měl umožnit evidovat různé druhy majetku, měly by být udržovány všechny záznamy o změnách a podobně. Taktéž by mělo být umožněno zaznamenávat případné revize majetku (typicky elektrických zařízení).
2. Systém by měl automaticky, nejlépe emailem, upozorňovat na nutnost opětovné evidence nebo revize.
3. Systém koncipujte tak, aby bylo možné evidovaný majetek označovat štítky s čárkovým kódem a stejně tak, aby bylo možné za pomoci čtečky čárového kódu majetek snadno určit.
4. Klientskou část systému koncipujte tak, aby bylo možné majetek evidovat jak z klasického PC tak například z PDA.
5. Systém dobře zdokumentujte.

Seznam doporučené odborné literatury:


Podle pokynů vedoucího bakalářské práce.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.


Vedoucí bakalářské práce: **Ing. David Seidl**

Datum zadání: 30.11.2008

Datum odevzdání: 07.05.2010


doc. Dr. Ing. Eduard Sojka
vedoucí katedry




prof. Ing. Ivo Vondrák, CSc.
děkan fakulty

Prohlášení studenta

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

Dne 26. 4. 2010

Lukáš Veselý

Poděkování

Děkuji vedoucímu bakalářské práce, Ing. Davidu Seidlovi, za ochotu pomoci a čas, který mé práci věnoval. Děkuji za cenné rady a připomínky, bez kterých by tato práce nevznikla.

Abstrakt

Práce se zabývá hlavně skriptovacím jazykem PHP, knihovnou FPdf pro online tvorbu PDF dokumentů a čárovými kódy. Popisuje využití zmíněné knihovny pro online generování čárových kódů. Je podrobně rozebrána problematika čárových kódů a jejich využití pro konkrétní řešení.

Praktickou část práce tvoří informační systém evidence majetku, na kterém jsou zjištěné poznatky demonstrovány. Jsou popsány jeho funkce, vlastnosti a způsob implementace.

Zmíněny jsou také jiné technologie využívané pro tvorbu webových informačních systémů, popřípadě pro tvorbu internetových stránek.

Abstract

The thesis goes in for the PHP scripting language, the FPdf library, on-line creation of portable documents (PDF) and bar codes. Usage of the stated library is described in relationship with the on-line bar code generator. The proper description of the bar code matters and usage in a particular implementation cannot be missing.

The practical part of the thesis is represented by an information system used for property agenda where a user is able to notice the researched findings along with a description of its functions, characteristics and a way how they are implemented.

There is also a reference to other technologies designated for development of web-based information systems, possibly web pages.

Klíčová slova

php, php injection, mysql, mysql injection, database, skripty, ajax, javascript, čárový kód, code 39, qr code, pdf, fpdf, automatizace, sběr dat

Keywords

php, php injection, mysql, mysql injection, database, scripts, ajax, javascript, barcode, code 39, qr code, pdf, fpdf, automatization, data capture

Obsah

1. ÚVOD.....	6
2. ČÁROVÉ KÓDY	7
2.1. OBECNĚ	7
2.2. SNÍMAČE	7
2.3. CODE 39 MOD 43	8
2.4. BUDOUCNOST.....	9
3. 2D KÓDY	10
3.1. OBECNĚ	10
3.2. SNÍMAČE	10
3.3. QR KÓDY	10
3.3. OSTATNÍ DRUHY 2D KÓDŮ	12
4. NÁVRH A TVORBA SYSTÉMU	12
4.1. OBECNĚ	12
4.2. NÁROKY NA PROVOZOVÁNÍ SYSTÉMU.....	12
4.3. UŽIVATELÉ A PRÁVA PRO PŘÍSTUP	13
4.4. VÝSTUP	14
4.5. POUŽITÉ TECHNOLOGIE.....	14
4.5.1. PHP	14
4.5.2. HTML	16
4.5.3. CSS	16
4.5.4. JAVASCRIPT	17
4.5.5. AJAX.....	17
4.5.6. MYSQL	18
4.6. IMPLEMENTACE.....	20
4.6.1. PŘÍKLAD VYTVOŘENÍ PDF DOKUMENTU ZA POMOCÍ KNIHOVNY FPdf	20
4.6.2. POPIS ALGORITMU PRO GENEROVÁNÍ CODE 39 MOD 43	21
4.6.3. ALGORITMUS PRO GENEROVÁNÍ QR CODE	23
4.7. UPOZORNĚNÍ NA REVIZI MAJETKU	24
4.8. BEZPEČNOST	24
4.9. EVIDENCE ZMĚN A VYŘAZENÉHO MAJETKU	26
4.10. UKÁZKA OVLÁDÁNÍ SYSTÉMU	28
4.11. VERZE PRO MOBILNÍ TELEFON	30
5. ROZŠÍŘITELNOST SYSTÉMU.....	32
6. ZÁVĚR.....	33
7. POUŽITÉ ZDROJE.....	34
8. PŘÍLOHY	36
8.1. PŘÍLOHA Č. 1 – DATOVÝ SLOVNÍK.....	36
8.2. PŘÍLOHA Č. 2 – ER DIAGRAM	37

1. Úvod

Počítače a internet jsou nedílnou součástí dnešní doby. Jednou z mnoha oblastí, která díky nim zaznamenala výrazný pokrok, je spravování a archivace dat. V této práci se zaměřuji na evidenci drobných a krátkodobých předmětů (dále jen majetku) s pomocí online informačního systému, schopného generovat čárové kódy a být zcela nezávislý na jiných aplikacích. Webová aplikace je dostupná na všech počítačích, noteboocích, mobilních telefonech a jiných zařízeních vybavených internetovým prohlížečem a připojením k internetu. Systém je nezávislý na konkrétním hardwaru či typu operačního systému a je možné k němu přistupovat snadno a rychle. Čárové kódy mají na rozdíl od jiných modernějších identifikačních prostředků (např. RF čipů) tu výhodu, že není-li k dispozici čtecí zařízení (typicky čtečka čárových kódů), může se zadat kód přímo na klávesnici daného zařízení. Další výhodou online řešení je schopnost systému upozorňovat E-mailem správce majetku na nutnost revize a jiné periodicky opakující se činnosti, které musí pravidelně vykonávat.

2. Čárové kódy

2.1. Obecně

Čárové kódy jsou prostředkem pro automatizovaný sběr dat. V současnosti jsou nejrozšířenějším systémem pro identifikaci, a to zejména položek v obchodech a zásilek v přepravních službách. Setkat se s čárovými kódy je možné stále častěji také v jiných odvětvích a činnostech (knihovny, identifikační karty, parkovací systémy...). První patent byl udělen již v polovině minulého století.

2.2. Snímače

Snímače čárových kódů (čtečky) lze rozdělit podle těchto kritérií:

- Možnosti připojení (USB, DIN, PS2, COM, ...),
- Provedení konstrukce (ruční s kabelem, ruční bezdrátové, pevně zabudované),
- Druhu snímače.

Starší technologií, ale dosud hojně využívanou, jsou **čtečky s CCD snímačem**. Pro osvětlení čárového kódu se používají LED diody. Kód musí být přiblížen do bezprostřední blízkosti čtečky a pozitivní identifikace je omezena maximální snímanou šířkou, obvykle od 60 do 80 mm. Omezena je také možnost číst částečně poškozené kódy nebo kódy na nerovném podkladu.



Obrázek 1 - Čtečka čárových kódů s CCD snímačem - Zdroj [21]

Modernější jsou **čtečky s laserovým paprskem**. Lze jimi číst kódy do větší vzdálenosti, cca 100 mm, s maximální šířkou až cca 150 mm. Paprsek je velmi tenký, v případě částečného poškození čárového kódu stačí souvislá nepoškozená sekvence linek alespoň o tloušťce 1 – 2 mm. Samotný čárový kód může být proto velmi úzký. Laserové snímače se dělí na čistě horizontální a na kombinaci snímání vertikálního a horizontálního. Takto pokročilý způsob se využívá u průmyslových čteček s množstvím zrcadel, které dokážou přečíst kód i libovolně natočený.



Obrázek 2 - Čtečka čárových kódů s laserovým snímačem - Zdroj [22]

Poslední možností, jak přečíst čárový kód, je analýza rastrového snímku, založená na podobném principu jako je OCR (*Optical Character Recognition*) u textu. Za pomoci vhodného softwaru lze přečíst i čárový kód vyfocený fotoaparátem v mobilním telefonu. Tuto možnost uvádím jen pro úplnost, jedná se o neefektivní a běžně nevyužívané řešení. Pro mobilní telefony je speciálně vyvíjen jiný typ kódu, a to tzv. QR. O tomto se ještě ve své práci dále zmíním.

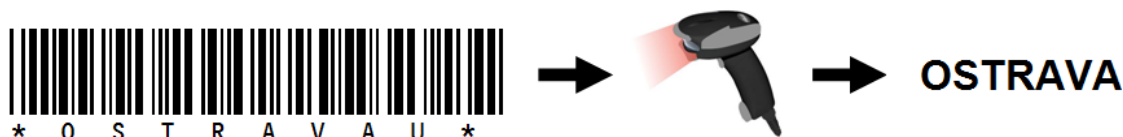
2.3. Code 39

Existuje mnoho typů čárových kódů. Běžný člověk se nejčastěji setká s kódy standardu EAN-13, jimi je označováno veškeré zboží v obchodech. Pro účel mého systému není příliš vhodný, neboť způsob kódování je složitý a kód se skládá z několika částí. Ocení ho zejména výrobci, mohou využít prefixů pro kódy zemí, druhy zboží atd.

Code 39 je standart, který jsem si vybral já. Z názvu plyne, že je schopen zakódovat 39 různých znaků (velká písmena (A až Z), čísla (0 až 9) a několik speciálních znaků). Využívá se hlavně v logistice a knihovnách. Každý znak je reprezentován pěti čarami a čtyřmi mezerami mezi nimi. Z těchto devíti prvků jsou vždy 3 široké a 6 úzkých. Malá písmena nejsou podporována a jsou na vstupu automaticky konvertována na velká. Znak "hvězdička" je vyhrazen pro znaky start a stop označující začátek a konec kódu pro čtecí zařízení.

Existuje také font písma podporující zapisování tohoto typu čárových kódů stejně jako běžného textu. Uživatel jen nesmí zapomenout, že řetězec vždy musí začínat a končit hvězdičkou. Jeden z fontů typu Code 39 se jmenuje „Free 3 of 9“.

Je možné se setkat i s označením Code 39 Mod 43. Mod 43 v názvu znamená, že kontrolní součet v kódu umístěn jako předposlední prvek je vypočítáván součtem všech hodnot znaků kódu modulu 43. Redukuje se tím možnost výskytu chyby.

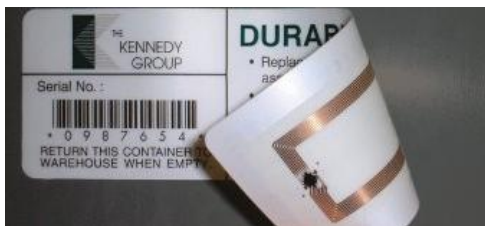


Obrázek 3 - Ukázka čárového kódu typu Code 39 mod 43

2.4. Budoucnost

Myslím si, že čárové kódy, i přes nové a modernější způsoby identifikace, budou používány i v budoucnu. Nejčastějším a očekávaným pokrokem bude zavedení bezkontaktních čipů na potraviny a spotřební zboží v běžných supermarketech. Zákazník přijde k pokladně, nebude muset čekat ve frontě a vykládat veškeré zboží na pás. Pokladní nebude muset každou věc přiložit ke snímači čárových kódů. Senzory u pokladny samy načtou informace z bezkontaktních čidel umístěných pod obalem každého výrobku. Zákazník bude odcházet od pokladny se zaplaceným nákupem mnohem dříve než dnes. Jakmile doma vyskládá potraviny do lednice, opět dojde k automatickému načtení informací o vloženém zboží. Lednice bude sama hlídat stav jeho domácích zásob. Někteří výrobci (Electrolux, Whirlpool či Philips) již uvedly na trh ledničky i s připojením k internetu, z objednávání potravin se tímto stává automatický proces.

To je zatím více či méně vzdálená budoucnost. Prvním krokem a hlavním důvodem pro zavedení této technologie bude rychlé odbavení zákazníka v obchodech. Zdá se, že po finanční stránce bude nejdražší pořízení zařízení na výrobu a identifikaci těchto bezkontaktních čidel. Opak je pravdou. Hlavním cílem bude minimalizovat náklady na výrobu bezkontaktních čidel samotných. Důvodem je, že obal výrobku bude po konzumaci či spotřebě vyhozen a s ním i ono čidlo, náklady na obal výrazně vzrostou. Ekonomicky výhodnějším řešením jsou proto stále čárové kódy.



Obrázek 4 - RFID pasivní čip - Zdroj [24]

3. 2D kódy

3.1. Obecně

Do svého systému jsem kromě čárových kódů zavedl i podporu tzv. 2D kódů. Často jsou chybně označovány také za kódy čárové, byť žádné čáry neobsahují a tím se od čárových kódů výrazně liší. Jejich využití je zejména v oblasti reklamy a všude tam kde je rychle potřeba předat větší množství informací a nechceme je ručně přepisovat do mobilního telefonu nebo počítače. Pro označení zboží, či v mém případě majetku, mají jednu velkou výhodu. Jsou malé a je možné je tak použít i na drobné elektronické součástky, na které by nalepení samolepky s velkým čárovým kódem nepřípadalo v úvahu.

3.2. Snímače

Pro profesionální využití se používají optické CMOS snímače. Na první pohled vypadají stejně jako snímače kódů čárových, rozdíl je ve výšce čtecího průzoru, která je vyšší. Takovéto čtečky dokážou přečíst kód na vzdálenost od půl centimetru až do 6 metrů. Při čtení z velké vzdálenosti je vyslán laserový paprsek pro zaměření kódu obsluhou. Lepší modely dokážou díky vestavěnému OCR přečíst i tištěný text.

Běžný uživatel však podobným zařízením nedisponuje a k přečtení 2D kódu použije svůj mobilní telefon s vhodným softwarem. Některé telefony značky Nokia mají podporu již z výroby, pro ostatní telefony je zde například software Kaywa Reader.

3.3. QR kódy

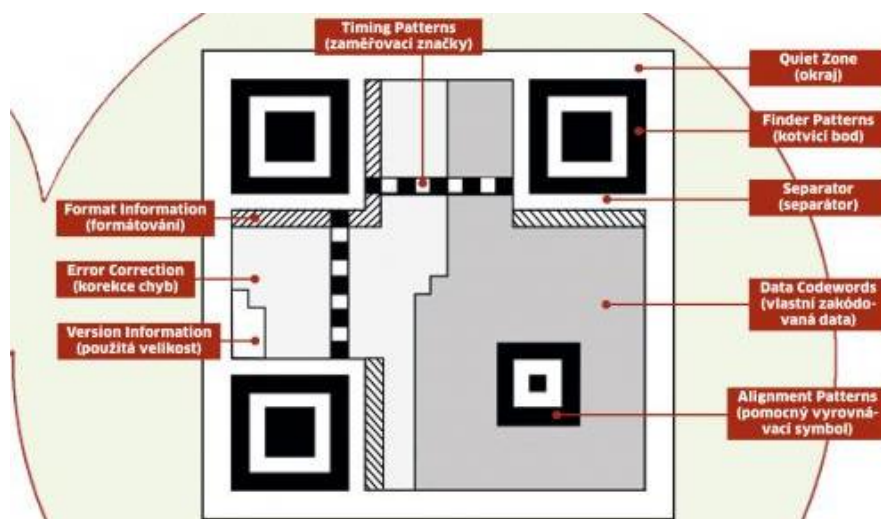
„Quick Response“ kódy (zkráceně označované jako QR kódy) jsem použil ve své práci já. Mnoho lidí u nás o nich doposud neslyšelo, i když v Asii se používají již delší dobu. Jejich primárním účelem je snadno přenést do mobilního telefonu URL odkaz. Člověk se může běžně v Japonsku setkat s QR kódy například na plakátech v kinech. Po sejmutí QR kódu se dostane na internetové stránky o filmu, kde může zdarma stáhnout vyzváněcí melodii nebo tapetu s tematikou filmu. Firma McDonald umístuje QR kódy na obaly od svých hamburgerů a zákazník se rychle dostane k nutričním hodnotám toho, co právě konzumuje. Výjimkou nejsou ani QR kódy na firemních vizitkách. Masovou podporu zahájila firma Google ve svých tištěných reklamách, snaží se propojit tištěnou reklamu s internetovou. Časem se i u nás QR kód stane běžnou záležitostí, podporu zahájil po vzoru Googlu i náš český Seznam.



Obrázek 5 - QR kód na reklamním plakátu společnosti Seznam - Zdroj [14]

Přenos URL odkazu není jedinou schopností tohoto kódu. Do QR kódu je možné zakódovat libovolný text a požívat ho i místo standardních čárových kódů. Konkurenceschopnost čárovému kódu je v jeho malých rozměrech.

Struktura kódu se skládá z matice označené jako velikost 1, která obsahuje 21 x 21 bodů až po matici označenou jako velikost 40 s obsahem 177 x 177 bodů. Do velikosti QR kódu 40 je možné zakódovat binárně až 4300 znaků. Konverze probíhá tak, že se text převede na posloupnost jedniček a nul. Jedničky jsou černá pole a nuly pole bílá. Přesný význam jednotlivých částí QR kódu je jasný z následujícího schématu:

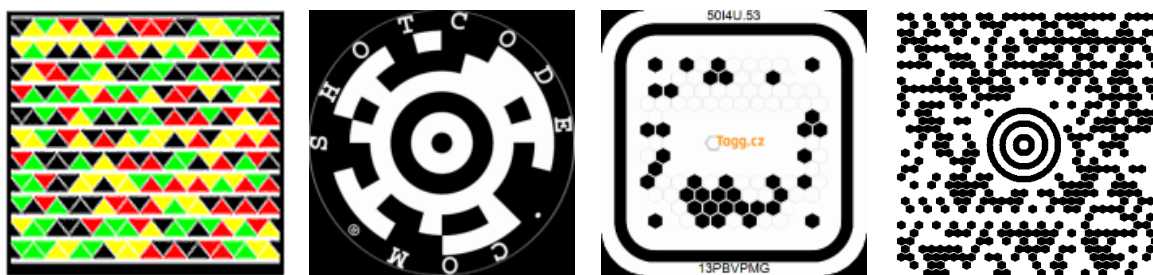


Obrázek 6 - Struktura QR kódu - Zdroj [8]

QR kód se může stejně jako čárový kód ušpinit nebo jinak poškodit. Inteligentní korekce chyb dokáže obnovit až 30% poškozených dat, záleží na konkrétním typu kódu. Tato možnost je kompenzována většími rozměry. U nejmenších QR kódů je možné obnovit pouze asi 7% poškozených dat. Ve svém systému používám velikost 1, označovanou také jako velikost L.

3.4. Ostatní druhy 2D kódů

Stejně jako existuje nepřeberné množství druhů kódů čárových, existuje i mnoho druhů 2D kódů. Mohu jmenovat například Data Matrix pro značení zejména integrovaných obvodů, velkokapacitní UPS Code využívaný poštou v USA nebo barevný Multi Color Barcode od Microsoftu. Při hledání informací k tomuto tématu jsem narazil až na cca 35 různých druhů. Existence řady standardů a různorodých licenčních podmínek je hlavní překážkou pro jejich masové rozšíření. Software a zařízení, které by dokázalo číst všechny existující 2D kódy, je zatím utopie.



Obrázek 7 - Multi Color Barcode, Shootcode, BeeTags a UPS code - Zdroj [27]

4. Návrh a tvorba systému

4.1. Obecně

Systém je koncipován jako přehledná, jednoduchá a uživatelsky přívětivá webová aplikace. Základem je skriptovací jazyk PHP. Jeho jednoduchost a univerzálnost umožnila integrovat generátor čárových kódů i QR kódů přímo do aplikace, bez nutnosti použít jiná externí řešení. Dalšími použitými technologiemi je značkovací jazyk HTML, kaskádové styly CSS, objektově orientovaný skriptovací jazyk JavaScript, technologie interaktivních webových aplikací AJAX, relační databáze MySQL a volně šiřitelná knihovna FPdf pro generování PDF souborů s pomocí PHP. Všechny uvedené technologie jsou vysvětleny v kapitole s názvem Použité technologie.

Do aplikace je možné přistupovat ze všech zařízení vybavených internetovým prohlížečem a připojením k Internetu. Uživatel není omezen pouze na osobní počítač nebo notebook, může využít i mobilní telefon, PDA a podobné zařízení. Systém pro tato zařízení má jen omezené funkce, využívá jiné, graficky jednodušší rozhraní. Pokud systém jednoznačně zjistí, že uživatel přistupuje do systému přes takovéto zařízení, bude automaticky na toto graficky jednodušší rozhraní přepnut. Možnost vrátit se ke standardnímu zobrazení nechybí. Druhé podstatné omezení u uvedených zařízení je v podpoře tisku.

4.2. Nároky na provozování systému

Pro fungování systému a snadný přístup k němu je nutné mít webový server s podporou PHP skriptů a MySQL databází připojený minimálně do místní sítě. Systém je pak dostupný pro všechny registrované uživatele z dané organizace či firmy, pokud jsou fyzicky připojeni do Intranetu. Daleko lepším řešením, umožňujícím přístup z jakéhokoliv místa, je webový server připojený přímo do Internetu. Hlavním kritériem volby řešení je výše požadované investice a další individuální hlediska každého subjektu - organizace či firmy. Pro vývoj a testovací provoz své bakalářské práce jsem vybral jinou, dosud neuvedenou, variantu. Systém jsem vyvíjel a testoval na zaplaceném hostingu od společnosti Gigaserver.cz. Je přístupný na [www](http://www.majetek.lucinkapes.com) adrese <http://majetek.lucinkapes.com>. Toto řešení není pro ostré nasazení ideální z důvodu:

- omezené možnosti nastavení konfigurace serveru,
- nedostupnosti systému při výpadku internetového připojení,
- omezené možnosti zálohování databáze.

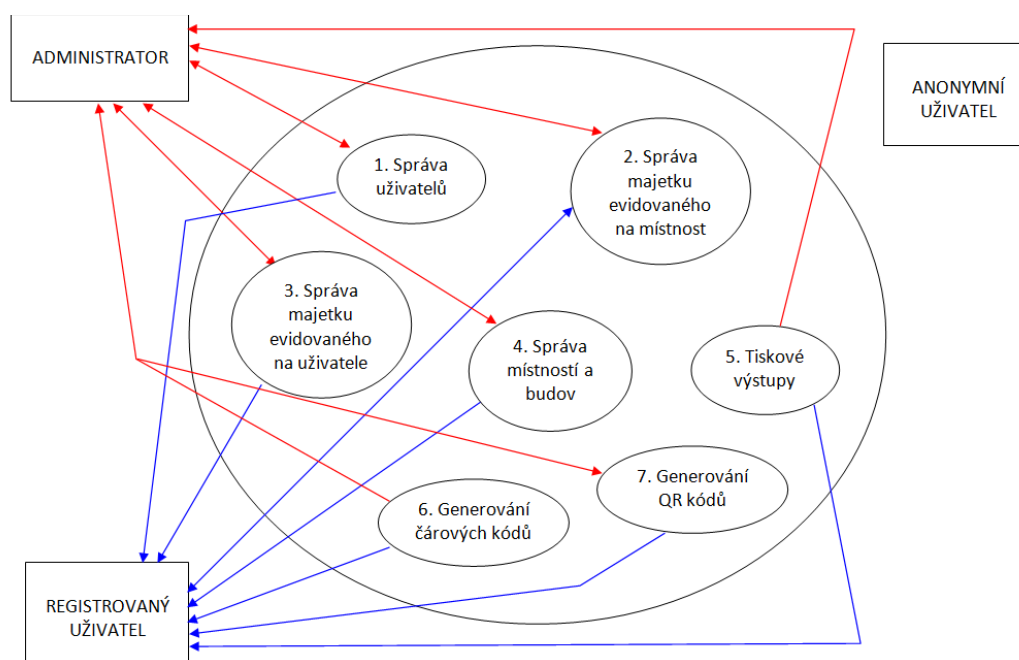
4.3. Uživatelé a práva pro přístup

Přístup do systému je povolen jen pro registrované a následně přihlášené uživatele. Tito uživatelé se dále dělí na dva druhy s odlišnými právy:

- **Běžný uživatel** (registrovaný uživatel) - správce místnosti nebo uživatel nějakého evidovaného majetku.
- **Administrátor** - uživatel obohacen o administrátorská práva.

Administrátorských účtů může být více, nejméně však jeden. Učinit z běžného uživatele administrátora může jen jiný administrátor. Stejně tak jen administrátor může přidávat účty běžných uživatelů. Do systému není umožněno registrovat se komukoliv a neoprávněně nahlížet na uložená data.

Rozdělení přístupových práv je patrné z následujícího diagramu datový toků označovaného anglickou zkratkou DFD (Data Flow Diagram).

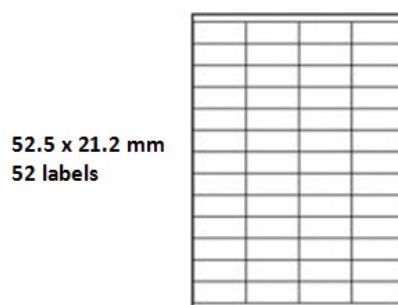


Obrázek 8 – DFD diagram

4.4. Výstup

Jedním z hlavních požadavků na aplikaci je tisk čárových kódů. Nejefektivnějším řešením, nabízejícím široké možnosti a nezávislost na platformě, je formát PDF. Za pomoci knihovny FPdf je možné do PDF vpisovat nejen text, ale i grafiku. Těchto možností jsem nevyužil pouze pro čárové kódy, ale také pro QR kódy a inventáře majetku v konkrétní místnosti. Důležitou vlastností pro tisk čárových kódů je stoprocentní kontrola nad jejich rozmístěním na stránce. Je naprosto zaručeno, že se při tisku vždy trefím do středu samolepících štítků v každém archu.

Systém je vytvořen pro tisk čárových kódů na samolepící etikety od firmy LOGO. Jedná se o samolepící papír formátu A4, na kterém je umístěno 52 etiket v matici 4 x 13. V případě potřeby není problém upravit zdrojové kódy pro tisk na archy s jiným počtem a velikostí štítků nebo i jiný formát papíru.



Obrázek 9 - Čtečka čárových kódů s CCD snímačem

QR kódy jsou na stránce rozmístěny v matici 10 x 3 a jejich rozmístění není odvozeno od žádného druhu archu se samolepícími štítky. Je možné tak tisknout kódy jen na běžný papír formátu A4. Úprava rozmístění by byla v případě ostrého nasazení provedena až na základě výběru konkrétních samolepících archů.

4.5. Použité technologie

4.5.1. PHP

PHP je skriptovací jazyk pro tvorbu dynamických webových aplikací, disponující řadou knihoven a internetových protokolů, vycházející s jazyků C, Perl, Java a dalších. PHP skripty se zapisují přímo do HTML kódu stránky a jsou prováděny na straně serveru. Syntaxe je poměrně volná a svobodomyslná. To je jeden z důvodů vysoké popularity tohoto jazyka a jeho rozšířené podpory ze strany hostingových firem. PHP soubory mají nejčastěji příponu `.php`. Používají se i přípony `.php3`, `.php4`, `.php5` a `.phtml`. Pokud vyjde nová verze PHP, bude celý web působit zastarale, a proto nejsou příliš doporučovány.

Pro PHP najde využití i začínající programátor, vytvářející pouze statické webové stránky. Nepotřebuje pracovat s databázemi, provádět výpočty, načítat data ze souborů či odesílat přes HTML formuláře E-maily. Každá běžná webová stránka se skládá ze záhlaví, menu, těla a zápatí. Většinou je potřeba při procházení takového webu měnit jen obsah těla. K tomuto úkolu je PHP ideální.

Častým problémem prvních PHP aplikací začínajících programátorů je otázka bezpečnosti. I pomocí jednoduchého kódu lze zlikvidovat celý obsah webu a je jen na programátorovi, jak moc na to bude při vývoji myslet. Nejčastěji se jedná o tzv. PHP injection a SQL injection.

Příkladem špatně napsaného a potencionálně nebezpečného zdrojového kódu, umožňujícího útok za pomoci **PHP injection**, je například tento:

```
<?php include $_GET["page"] ?>
```

Napsat vložení obsahu jiného souboru tímto způsobem a použít URL s parametry v následujícím tvaru je jednoduché.

```
index.php/page=novinky.php
```

Není-li přímo v nastavení serveru zakázáno `allow_url_include`, může útočník zmanipulovat parametr v URL a naincludovat do původního skriptu svůj vlastní nebezpečný kód umístěný na jiné doméně.

index.php/page=http://domena.com/nebezpecny_kod.php

Při vývoji mého systému jsem neměl možnost přímo ovlivnit nastavení serveru. Tato skutečnost je běžná na většině placených i neplacených hostingových serverech. Svůj kód jsem upravil tak, aby útok pomocí PHP injection nebyl možný. Přidáním tečky a lomítka před název souboru je znemožněno vložit soubor uložený mimo adresář na serveru.

Soubor: /index.php

```
52      {
53          $inc = "../".$page.".php";
54      }
55      include($inc);
56 ?>
```

Stejného efektu se dosáhne funkcí `is_file()`. Ta vrátí `false` v případě, že se daný soubor nenachází na serveru. Účinnou a poslední obranou, kterou je možné použít, je konstrukce `switch()`.

```
<?php
    switch($_GET["page"])
    {
        case "novinky":
            include "novinky.php";
            break;
        default:
            include "404.php";
            break;
    }
?>
```

Druhou hrozbou je **SQL injection**. Typickým příkladem je špatné napsání přihlašovacího skriptu. Odešle-li se přihlašovací formulář na PHP script, který bez dalších úprav ověřuje zadané jméno a heslo tímto způsobem

```
"SELECT * FROM uzivatele
WHERE jmeno = '". $j. "'" AND heslo = '". $h. "'"
```

je možné, aby útočník jako jméno a heslo zadal `' OR 1=1' --`. Vznikne dotaz, který obejde autorizační proceduru, neboť červená část bude vnímána jako komentář.

```
"SELECT * FROM uzivatele
WHERE jmeno = '". $j. "'" OR 1=1" --' AND' heslo='". $h. "'"
```

Horší situace nastane, odhadne-li útočník název tabulek v databázi. Celé je může odstranit. Obrana proti SQL injection se dělí podle místa, kde se s ní bojuje. Je to na straně databáze nebo na straně aplikace. Boj na straně databáze pomocí vhodného nastavení přístupových práv pro

aplikační vrstvu je jednoduchý a dokáže zabránit odstranění celé tabulky nebo dokonce databáze. Nevyřeší ale všechny problémy. Účinnější je boj na straně aplikace. Použil jsem vestavěnou funkci PHP `mysql_escape_string()`, která nebezpečné části dotazu opatří zpětným lomítkem. Alternativou je funkce `mysql_real_escape_string()`. Rozdíl mezi oběma funkcemi je, že `mysql_real_escape_string()` bere v úvahu i aktuálně nastavenou znakovou sadu spojení a má tím o jeden parametr navíc. Ukázka ošetřeného SQL dotazu z mé aplikace:

Soubor: /script.php

```
40 $login = mysql_escape_string($_POST['username']);
41 $heslo = MD5(mysql_escape_string($_POST['password']));
42
43 $dotaz = mysql_query("SELECT * FROM majetek_uzivatele
44                      WHERE login = '$login' AND heslo = '$heslo'",
```

Při vývoji mého systému jsem využil v PHP zejména vestavěné knihovny pro práci s MySQL databází a okrajově pro práci s XML soubory. Externí použitou knihovnou byla knihovna FPdf.

4.5.2. HTML

HyperText Markup Language je značkový jazyk umožňující publikaci dokumentů v systému World Wide Web. Zapisuje se párovými a nepárovými tagy. HTML dokument musí dodržet správnou strukturu. Historie jazyka sahá k začátkům devadesátých let minulého století. Hlavní vize při jeho vývoji byla možnost odkazovat se z jednoho dokumentu na jiný a vytvořit křížové odkazy. V současnosti se již používá verze 5. Tato technologie je natolik známá, že její další popis v rámci této práce není nutný.

4.5.3. CSS

Layout prezentační vrstvy systému je vytvořen strukturou tabulek, které jsou jako ostatní HTML elementy naformátovány pomocí kaskádových stylů (CSS - Cascading Style Sheets).

Soubor s kaskádovými styly, na které se odvolávám pomocí názvů tříd, id elementů nebo názvů tagů, je pojmenován `style.css`. Tato koncepce umožňuje v případě potřeby upravovat hromadně a snadno vzhled celé aplikace, je zde i možnost tvorby skinů. V samostatného souboru `style_menu.css` je stylopis pro základní menu informačního systému. Díky němu vypadá jako plastické záložky (ouška listů).

4.5.4. JavaScript

JavaScript se na rozdíl od PHP provádí na straně klienta. Stejně jako PHP vychází z jazyků C nebo Java. S Javou ale nemá kromě názvu a podobné syntaxe nic společného, toto pojmenování je jen marketingový tah. Podmínkou fungování JavaScriptu je podpora ze strany internetového prohlížeče, je třeba mít v nastavení prohlížeče JavaScript povolen. V minulosti to nebylo tak samozřejmé jako dnes.

V mojí aplikaci jsou mnou psané JavaScripty v souboru `javascript.js`. Tento soubor obsahuje funkce pro kontrolu smysluplnosti vstupních dat, které zadává uživatel do formulářů, zajišťující automatické odhlášení, přidávající dynamicky pole do formuláře, zobrazení řadících tlačítek u tabulkových výpisů, detekce prohlížeče (zařízení uživatele) a další.

Dále využívám JavaScript pro zlepšení uživatelského komfortu grafického rozhraní systému. Pomocí události `onclick()` jsou zobrazovány určité části stránky, které jsou pro přehlednost defaultně skryté. Z technologie JavaScriptu vychází i další použitá technologie, a tou je AJAX.

4.5.5. AJAX

AJAX (Asynchronous JavaScript and XML) využívá kromě JavaScriptu i XML. Nejsilnější stránkou AJAXu je možnost vytvářet interaktivní webové aplikace bez nutnosti neustále znovu načítat stránku. Mění se jen obsah té části stránky, kde je to třeba. Hlavní výhodou je zároveň nejčastěji zmiňovaná nevýhoda. Z webové stránky se stává plnohodnotná aplikace s vnitřní logikou a ne jenom pouhá posloupnost na sebe navazujících dokumentů s hypertextovými odkazy. Není možné se mezi stránkami navigovat pomocí tlačítek Zpět a Vpřed. To lze do jisté míry kompenzovat zásahem programátora. Narůstá přitom ale složitost aplikační logiky. Další výhodou je nižší zátěž serveru, neboť není nutné sestavovat celý HTML dokument. Při špatné implementaci se tento fakt nemusí vždy podařit a stoupá počet HTTP požadavků, i když s menší datovou velikostí. Posledním drobným problémem, se kterým se setkávám jako běžný uživatel některých AJAXových aplikací, je absence signalizace práce na pozadí. Neimplementuje-li programátor nějaký signál (například animované přesýpací hodiny), uživatel během načítání změny AJAXové komponenty neví, zda se právě něco děje a vše funguje správně. Prohlížeč sám práci AJAXových komponent nijak neindikuje. Uživatel může nabýt dojmu, že mu přestalo fungovat internetové připojení nebo se stala jiná chyba a snaží se stránku obnovit. Tím přeruší práci AJAXové komponenty a nedosáhne očekávaného výsledku.

Celá technologie je relativně mladá, první zmínky o ní se objevily v roce 2005. I přes uvedené nevýhody je AJAX výborná technologie, která má před sebou další vývoj a budoucnost. Stále častěji se i složité aplikace vyvíjejí jako online www stránky.

Majetek v mém systému, jak je již uvedeno, je možné evidovat na konkrétního uživatele nebo na místnost. AJAX využívám pro evidenci na místnost a možnost si ji zvolit v návaznosti na konkrétní budovu. Uživatel si v rozevíracím seznamu vybere budovu a ve vedlejším seznamu se dynamicky načte seznam místností, které se v ní nacházejí. Skript funkce je uložen v souboru `chainedselects.js`. Data pro rozevírací seznamy jsou získávána z XML souboru `csconfig.php`. XML soubor má neobvyklou příponu PHP. Je to způsobeno tím, že jeho XML obsah se při každém použití sestavuje na základě dat vyselektovaných z MySQL databáze.

4.5.6. MySQL

MySQL je populární databázový systém vytvořený ve Švédsku, který tvoří s PHP oblíbenou dvojici. Stejně jako PHP je podporován širokým spektrem hostingových firem. Pro dotazování do databáze se využívá, jak již z názvu vyplývá, jazyk SQL (Structured Query Language), stejně jako do jiných SQL databázových systémů (např. PostgreSQL, MS-Sql). Funkce MySQL dostupné dnes, nebyly v minulosti samozřejmostí. Teprve v posledních letech bylo implementováno vytváření pohledů či triggerů.

Obsluhu MySQL databáze, uložené na vzdáleném serveru, je možné kromě vlastních skriptů provádět za pomoci PHP volně dostupného webového rozhraní phpMyAdmin. Jiný nástroj pro správu MySQL databází, rozšířený v takové míře a tolika jazykových mutacích, jako je právě phpMyAdmin, není jednoduché najít. Umožňuje provádět veškeré operace s MySQL databázemi, od vytváření samotných databází a tabulek až po zálohování (export) a obnovu (import) dat. Uživatel si sám zvolí, zda bude potřebnou operaci provádět přes grafické rozhraní a postupně se jím prokliká, nebo sám napíše SQL dotaz a nechá ho jen vykonat.

Moje databáze obsahuje celkem sedm vzájemně propojených tabulek. Datově nejobjemnější jsou ty na evidenci majetku, které se na data z ostatních tabulek (například místností a uživatele) odkazují pomocí cizích klíčů.

Seznam všech tabulek spolu s jejich entitami je uveden v datovém slovníku, který je *přílohou číslo 1* této bakalářské práce. Provázanost mezi tabulkami a integritní omezení jsou nejlépe patrná z entitně relačního diagramu (ER-diagram) – *příloha číslo 2*.

Databázi MySQL jsem vybral pro svou aplikaci z důvodu jejího rozšíření mezi hostingovými poskytovateli, nezávislosti na platformě a bezplatnému použití databáze samotné i administračního rozhraní PHP MyAdmin.

4.6. Implementace

V počátcích tvorby aplikace jsem zvažoval nejlepší řešení pro tvorbu čárových kódů. První mne napadlo generovat čárové kódy jako JPEG nebo GIF rastrové obrázky, které následně zpracuji do PDF dokumentu pro tisk. Z části jsem toto řešení měl i naprogramováno pro čárové kódy typu EAN 13. Řešení to bylo funkční, ale zbytečně náročné. V případě, že uživatel generoval několik set kódů, znamenalo to vytvořit a uložit několik set rastrových obrázků a všechny je vložit do PDF.

Změnu celého postupu za efektivnější jsem provedl, až jsem se začal zajímat o tvorbu PDF v PHP. Objevil jsem projekt PDFlib, který však je komerční, a pro mne tedy nevhodný. Následně jsem našel volně šiřitelnou (tzv. opensource) knihovnu FPDF, která umožňuje psát a kreslit přímo do PDF dokumentu při jeho tvorbě. Umožňuje za pomoci PHP příkazů vytvořit a zapsat například barevný a různě naformátovaný text, tabulky, odkazy i jednoduché grafiky (pro můj systém potřebné čárové a následně i QR kódy). Na domovských stránkách www.fpdf.org je podrobný manuál, vyžaduje ale dobrou znalost jazyka PHP.

4.6.1. Příklad vytvoření PDF dokumentu za pomoci knihovny FPDF

Základem je vložit si do PHP scriptu script knihovny FPDF. Toto zajistí funkce `require()`. Následuje vytvoření objektu (instance) třídy FPDF, přes tento objekt se přistupuje k jejím vestavěným funkcím. Pomocí `AddPage()` se přidá nová stránka do budoucího PDF dokumentu.

Soubor: /barcode2pdf.php

```
3 require('code39.class.php');
4 require('function.php');
5
6 $from = $_GET['from'];
7 $pdf=new PDF_Code39();
8 $pdf->AddPage(P,A4);
```

Funkce `SetFont()` nastaví font a jeho formátování. Prvním parametrem je název fontu. Je možné použít základní fonty (například Arial, Times, Courier, ...). nebo jiný (i vlastní) atypický font. Lze ho přidat pomocí funkce `AddFont()`. Dalším parametrem v pořadí je styl (B - tučné, I - kurziva, U – podtrženo). Při jeho vynechání se použije standardní styl. Poslední parametr udává velikost písma. Jednotka je stejná jako například v DTP Microsoft Word, a to typografický bod (anglická zkratka pt). Vložení textu je provedeno funkcí `Text()`, která podle zadaných parametrů vypíše na určitou pozici (souřadnicový systém xy, kde jednotkami jsou pixely) zadaný text.

Soubor: /code39.class.php

```
56 $this->SetFont('Arial','',10);
57 $this->Text($xpos, $ypos + $height + 4, "KOD: ".$code);
58 $this->SetFont('Arial','',7);
59 $this->Text($xpos, $ypos + $height - 9, "DKP VSB-TUO");
```

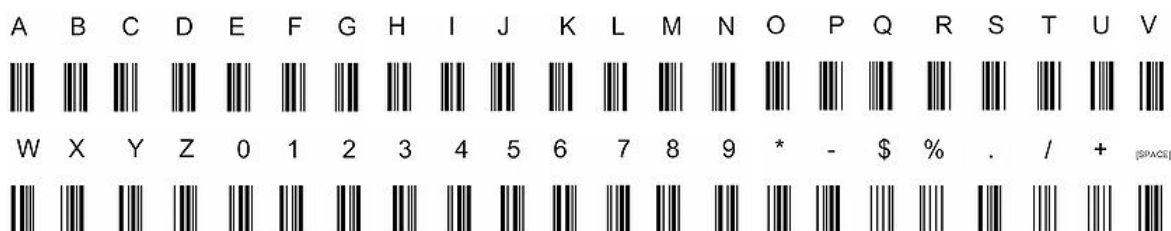
Na samotný závěr uvedeného ukázkového příkladu je volána funkce Output(). Parametry nejsou povinné a funkce defaultně odesílá vytvořený PDF dokument do internetového prohlížeče. Zadáním parametru "D" (download) nebo "F" (file) je uživateli nabídnuta možnost stáhnout dokument pod určitým názvem nebo ho jen uložit do adresáře na serveru.

Soubor: /barcode2pdf.php

```
95 $pdf->Output($name.'.pdf','D');
96 ?>
```

4.6.2. Popis algoritmu pro generování Code 39

Z informací o čárovém kódu typu Code 39, uvedených v jiné kapitole této práce vyplývá, že pro vytvoření algoritmu generujícího čárové kódy tohoto typu je třeba znát zápis uvedených 39 znaků:



Obrázek 10 - Čárové kódy typu Code 39 pro zakódovatelné znaky

Informace z obrázku č. 9 mi ještě neumožnily začít programovat. Posloupnost tenkých a tlustých linek či mezer jsem si převedl na posloupnost znaků. Sestavil jsem tabulku, s jejíž pomocí bylo zapsání kódu s vhodnou funkcí z knihovny FPdf poměrně jednoduchou záležitostí.

Znak	Vzor bsbsbsbsb	Znak	Vzor bsbsbsbsb	Znak	Vzor bsbsbsbsb	Znak	Vzor
0	nnnwwnwnn	C	wnwnnwnnn	O	wnnnwnnwn	-	nwnnnnwwn
1	wnnwnnnnw	D	nnnnwnnw	P	nnwnwnnw	.	wwnnnwnn
2	nnwwnnnnw	E	wnnnwnnn	Q	nnnnnnwww	SP	nwwnnnwnn
3	wnwwnnnnn	F	nnwnwnnn	R	wnnnnnwwn	*	nwnnwnnn
4	nnnwwnnnw	G	nnnnnwwnw	S	nnwnnwwn	\$	nwnwnwnn
5	wnnwnnnnn	H	wnnnnnwwn	T	nnnnwnwn	/	nwnwnnwwn
6	nnwwnnnnn	I	nnwnnwwn	U	wwnnnnnw	+	nwnnwnnw
7	nnnwnnwwn	J	nnnnwwwwn	V	nwwnnnnnw	%	nnnwnnwwn
8	wnnwnnwnn	K	wnnnnnnww	W	wwwnnnnnn		
9	nnwwnnwnn	L	nnwnnnnw	X	nwnnwnnnw		

A	wnnnnwnnw	M	wnwnnnwn	Y	wwnnwnnnn		
B	nnwnwnnw	N	nnnnwnnw	Z	nwnwnnnn		

Obrázek 11 - Tabulka zakódování znaků podle tloušťky linek

Algoritmus použitý při implementaci v praktické části bakalářské práce bere v úvahu možnost, že vstupními daty jsou malá písmena, která je nutné převést na velká a doplnit start a stop znak (hvězdičku). Nad čárový kód je zapsán text “DKP VSB-TUO“, pod čárový kód jeho textová podoba pro možnost identifikace bez čtečky čárových kódů.

Podle výše uvedené tabulky jsem vytvořil asociativní pole, kde indexem je jeden ze znaků, který je možné v Code 39 zakódovat. Hodnotou je kombinace písmen **n** (narrow) a **w** (wide). Na základě této hodnoty se vykreslí jednotlivé linky čárového kódu pro daný znak.

Soubor: /barcode2pdf.php

```

11 $barChar['0'] = 'nnnwwnwnn';
12 $barChar['1'] = 'wnnnwnnnw';
13 $barChar['2'] = 'nnwnnnnnw';
14 $barChar['3'] = 'wnwnnnnnn';
15 $barChar['4'] = 'nnnwwnnnw';
16 $barChar['5'] = 'wnnwwnnnn';
17 $barChar['6'] = 'nnwnwnnnn';
18 $barChar['7'] = 'nnnwnnwnw';
19 $barChar['8'] = 'wnnwnnwnn';
20 $barChar['9'] = 'nnwnnnwnn';
21 $barChar['A'] = 'wnnnwnnnw';
22 $barChar['B'] = 'nnwnwnnnw';

```

Hodnota čárového kódu, který je potřeba zakódovat, je uložena v proměnné `$code`. Je nutné před a za tento řetězec přidat znak `*` (tzv. start stop bit), označující začátek a konec kódu.

Je zahájen cyklus procházející postupně celý kód. Do proměnné `$char` jsou ukládány jednotlivé znaky kódu.

Je nutné ověřit, zda je daný znak možné zakódovat. To znamená zjistit, nachází-li se v poli `$barChar` pro něho vhodná sekvence tloušťky linek. Pokud je tato podmínka vyhodnocena pozitivně, je vypsáno chybové hlášení.

Sekvence tloušťky linek pro daný znak se uloží do pole `$seq`.

```

62 $code = '*' . strtoupper($code) . '*';
63 for($i=0; $i<strlen($code); $i++){
64     $char = $code[$i];
65     if(!isset($barChar[$char])){
66         $this->Error('Invalid character in barcode: ' . $char);
67     }
68     $seq = $barChar[$char];

```

Pole je postupně procházeno a proměnné `$lineWidth` je v každém cyklu přiřazena hodnota proměnné `$wide` nebo `$narrow` v závislosti na zakódování tloušťky linky (**w** nebo **n**). Hodnota proměnné `$wide` je nastavena programátorem. Pro správnou funkci čárového kódu je nutné hodnotu proměnné `$narrow` nastavit na jednu třetinu hodnoty proměnné `$wide`.

```
69         for($bar=0; $bar<9; $bar++){
70             if($seq{$bar} == 'n'){
71                 $lineWidth = $narrow;
72             }else{
73                 $lineWidth = $wide;
74             }
```

Při práci s každým lichým znakem čárového kódu (sudý index pole se znaky čárového kódu) je volána funkce `Rect()` z knihovny `FPdf`, která vytvoří obdélník (jednu linku čárového kódu). Parametry znamenají - zleva: horizontální pozice levého horního kraje obdélníku, vertikální pozice levého horního okraje obdélníku, šířka obdélníku, výška obdélníku a styl vykreslení (**F** = fill, obdélník bude vyplněný). Sudé znaky (liché indexy) nejsou vykreslovány, neboť jde o mezery mezi linkami.

```
75         if($bar % 2 == 0){
76             $this->Rect($xpos, $ypos, $lineWidth, $height, 'F'
77         }
```

Horizontální pozice je zvýšena o tloušťku nakresleného obdélníku nebo nenakreslené mezery. Cyklus pro vykreslování linek daného znaku čárového kódu je ukončen.

Následně je horizontální pozice pro vykreslování první linky dalšího znaku zvýšena o šířku tenké linky. Cyklus pro čárový kód je ukončen.

```
77             $xpos += $lineWidth;
78         }
79         $xpos += $narrow;
80     }
```

4.6.3. Algoritmus pro generování QR code

Tento algoritmus byl převzat z francouzského webu <http://prgm.spipu.net>. Autorem zdrojového kódu je Y. Swetake, řešení bylo uvolněno pod licencí GPL. Pracoval jsem na integraci algoritmu do mé aplikace, zejména pak na vhodném rozmístění QR kódů na stránce v požadovaném počtu a velikosti.

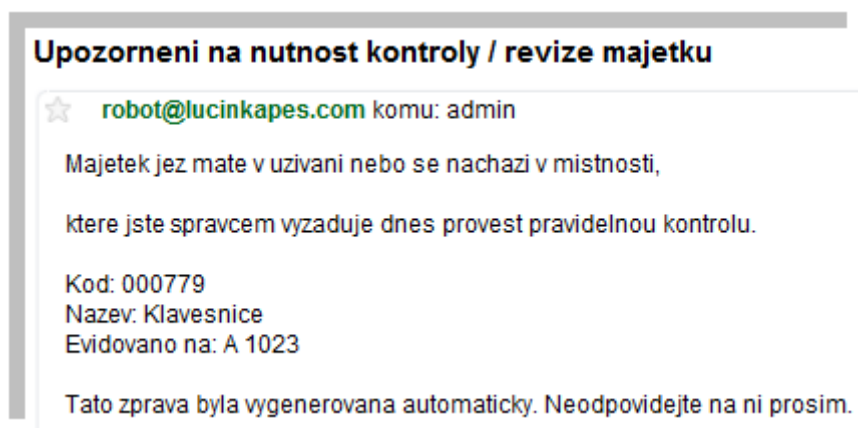
4.7. Upozornění na revizi majetku

Některý majetek je nutné periodicky kontrolovat (provádět revize). Typickým příkladem jsou elektrická zařízení nebo hasicí přístroje. Systém toto řeší zasíláním zpráv na E-mail:

- správce místnosti, kde se daná věc nachází,
- uživatele dané věci.

E-maily jsou zasílány pomocí PHP skriptu v souboru `cron.php`. O jeho pravidelné spouštění jednou denně, a to minutu po půlnoci, se stará tzv. CRON. Jedná se o automatizovaný spouštěč skriptů podle předem daných pravidel. Skripty je možné spouštět periodicky nebo v určitém datu a čase.

PHP skript se napojí na MySQL databázi. Vyhledá všechny evidované předměty, které mají datum příští upomínky nastavené na datum aktuálního dne v okamžiku provádění skriptu. Po odeslání E-mailu odpovědné osobě je v databázi datum příští upomínky přepsáno. Nové datum je aktuální datum, zvýšené o počet měsíců tvořících periodu upomínky. Uživatel obdrží E-mail podobný tomuto:



Obrázek 12 - Informační email s upozorněním na nutnost revize

4.8. Bezpečnost

Základní bezpečnostní hrozby vyplývající z použití PHP a MySQL databáze a jejich ošetření jsou již uvedeny v kapitole o použitých technologiích

Narušení bezpečnosti není vždy nutně způsobeno chybou programátora. Samotný uživatel může zapříčinit vlastní neopatrností zneužití svého účtu. Nejde jen o používání primitivního hesla nebo zaznamenání si hesla na lísteček umístěný vedle počítače. Stačí, aby uživatel opustil svůj

počítač, na kterém je stále přihlášený. Jednou z možností, jak v tomto může programátor pomoci, je nastavení doby expirace SESSION v případě, že se s aplikací nepracuje. Toto se dá nastavit v konfiguračním souboru `php.ini` u direktivy `session.cache_expire` pokud systém využívá vlastní server a programátor má plnou kontrolu nad jeho nastavením. U mého systému tomu tak není, a proto jsem se musel zamyslet nad jiným řešením. Implementoval jsem za pomoci JavaScriptu do hlavičky stránky časový odpočet zobrazující uživateli pomocí časové osy dobu zbývajících do automatického odhlášení. Tato doba je nastavena na 10 minut nečinnosti systému. Obnovuje se s každým novým požadavkem (znovu načtením stránky) nebo po kliknutí na tlačítko refresh za časovou osou. Blíží-li se časový odpočet do konce, je uživatel vizuálně upozorněn změnou barvy časové osy ze šedé na červenou.



Obrázek 13 - Vizuální zobrazení bezpečnostního limitu

Hlavní stránka `index.php` je sestavena mimo jiné ze stránky `head.php`, ve které je pomocí PHP scriptu vypsán následující JavaScript, je-li přihlášený nějaký uživatel. Funkce `CreateSecurityLimit()` se nachází jako ostatní JavaScriptový kód v souboru `javascript.js`. Aby se tato funkce opakovaně volala v intervalu jedna sekunda (1000 mili sekund) a docházelo k neustálému obnovování časové osy, využil jsem vestavěnou funkci `setInterval()`, které jsem jako parametr předal název své funkce.

Soubor: `/head.php`

```
18 <script>
19   CreateSecurityLimit();
20   window.setInterval("CreateSecurityLimit()", 1000);
21 </script>
```

Funkce `CreateSecurityLimit()` pracuje s globálně deklarovanou proměnnou `progressBarWidth`, která je uvedena na začátku souboru. V proměnné je uchována šířka časové osy v pixelech. Kód je pro přehlednost zapsán zkráceně a je vynecháno i přebarvení časové osy načerveno v případě, že zbývají poslední dvě minuty.

Soubor: `/javascript.js`

```
1 var progressBarWidth = 150;
12 function CreateSecurityLimit() {
```

Aby celková doba bezpečnostního intervalu byla 10 minut, je odečítáno od celkové délky 150 px pokaždé 0.25px ($(150 / 0.25 = 600) / 60 = 10$) tak dlouho, než je šířka časové osy 0 px (hodnota proměnné `progressBarWidth` je 0). Po jejím uplynutí je uživatel přesměrován na stránku `script.php` s parametrem v URL `?action=automatic_logout`.

```

13 progressBarWidth -= 0.25;
14 if (progressBarWidth > 0) {
15     document.getElementById("pb_dynamic").style.width = Math.round(progressBarWidth)+'px';
16     if(progressBarWidth==30) {
17         document.getElementById("pb_dynamic").style.backgroundColor = 'red';
18     }
19 }else{
20     window.location="script.php?action=automatic_logout";
21 }
22 }

```

PHP script v souboru `script.php` na základě parametru provede zničení SESSION. Tím se zničí informace o tom, že je uživatel do systému přihlášen a přesměruje ho na hlavní stránku (tj. přihlašovací formulář).

Soubor: `/script.php`

```

9 switch($action_get) {
10     case 'logout':
11         session_destroy();
12         header("location:index.php");
13     break;

```

4.9. Evidence změn a vyřazeného majetku

Jedním z požadavků na funkce systému bylo evidovat změny spojené s evidencí majetku. To zaručí dohledatelnost předchozích uživatelů, změny v názvu označení majetku, změny v umístění aj. informace.

Můj systém toto řeší pomocí dvou dalších tabulek. Jedné na evidenci změn pojmenovanou jako `majetek_editace`, která má strukturu nepatrně odlišnou od hlavní tabulky majetek. Druhá tabulka - na evidenci vyřazeného majetku - s názvem `majetek_vyrazeny`, má strukturu totožnou s hlavní tabulkou.

Údaje o editaci jsou zaznamenávány zapsáním aktuálních informací o editovaném majetku do tabulky `majetek_editace` ještě než dojde k samotné úpravě dat v tabulce majetek. Záznam je doplněn o datum provedení úpravy. V samotném grafickém rozhraní systému je možné tyto informace zobrazit kliknutím na tlačítko "historie" u výpisu majetku. Toto tlačítko je však aktivní jen v případě, že jsou pro danou položku evidovány nějaké změny.

Kód ▲▼	Název ▲▼	Datum přidání ▲▼	Přifazeno ▲▼	Upomínka (?) ▲▼	Příští upomínka			
000985	LCD monitor Benq 17"	2010-03-30 23:24:42	B 327	-	-	Upravit	Historie	Vyřadit

Kód ▲▼	Název ▲▼	Datum přidání ▲▼	Přifazeno ▲▼	Upomínka (?) ▲▼	Příští upomínka			
000985	LCD monitor Benq 17	2010-03-30 23:24:42	Hala 2	12	2011-03-30	Upravit	Historie	Vyřadit

Datum editace	Název	Přifazeno	Upomínka (?)
2010-03-30 23:26:39	LCD monitor Benq 17"	B 327	-

Obrázek 14 - Ukázka provedení editace (původní stav, nový stav, historie změn)

Vyřazení majetku neznamená, že bude ze systému úplně vymazán. Dojde jen k jeho přesunutí z tabulky majetek do tabulky majetek_vyrazeny. Tato operace proběhne díky shodné struktuře obou tabulek velmi jednoduše. Následně je pak ještě v tabulce majetek_vyrazeny do sloupce vyrazeno nastaveno aktuální datum.

Soubor: /script.php

```
23  mysql_query("INSERT INTO majetek_vyrazeny SELECT * FROM majetek WHERE barcode =  
24  mysql_query("UPDATE majetek_vyrazeny SET vyrazeno = '$datum_cas' WHERE barcode =  
25  mysql_query("DELETE FROM majetek WHERE barcode = '$barcode'", $pripojeni);
```

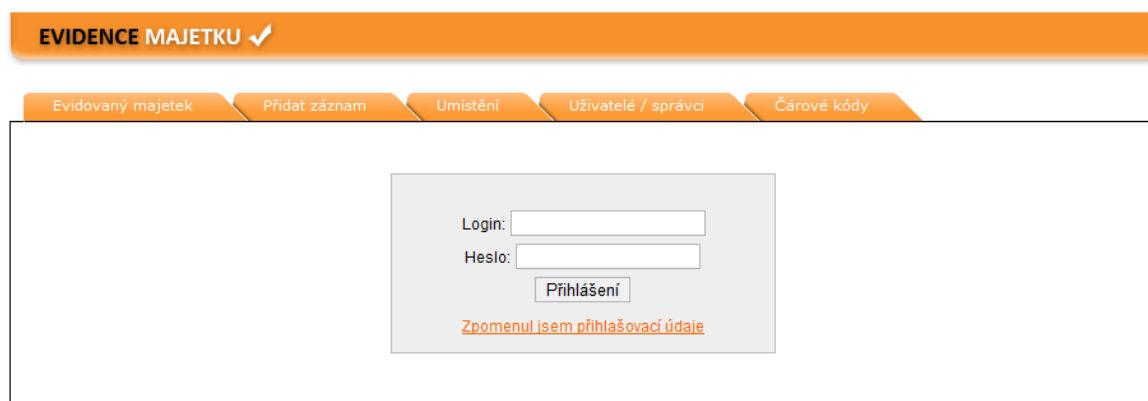
Vypsát veškerý vyřazený majetek je možné v systému pomocí jednoho z druhů výpisů v záložce „Evidovaný majetek“. Tento majetek již není možné jakkoliv editovat.

4.10. Ukázka ovládání systému

Následující posloupností kroků se uživatel přihlásí do systému, nechá si vypsát všechny evidovaný majetek a na závěr vygenerovat novou sérii čárových kódů.

- **Krok 1** – Uživatel se přihlásí do systému za pomoci svého uživatelského jména a hesla. Ukázka předpokládá přihlášení administrátora.

Zapomene-li uživatel své heslo, pokračuje odkazem “Zapomenul jsem přihlašovací údaje“, a na následující stránce si nechá vygenerovat heslo nové. Systém mu ho zašle na E-mail uvedený při registraci.



© 2009 Lukáš Veselý

Obrázek 13 - Přihlášení do systému

- **Krok 2** – V sekci Evidovaný majetek si vybere možnost “Chci vypsát všechny evidovaný majetek“. Dostane se na stránku s přehledným výpisem. Výpis lze libovolně řadit vzestupně i sestupně podle každé entity. Jako administrátor má právo měnit i mazat jednotlivé záznamy. (Běžný uživatel má právo upravovat a mazat jen majetek evidovaný na místnost, ve které je správcem).



© 2009 Lukáš Veselý

Obrázek 15 - Menu v sekci Evidovaný majetek

EVIDENCE MAJETKU

Bezpečnostní limit:

Evidovaný majetek

Přidat záznam

Umístění

Uživatelé / správci

Čárové kódy

Odhlásit

Aktuální pozice: evidovaný majetek - kompletní výpis

Kód ▲▼	Název ▲▼	Datum přidání ▲▼	Přiřazeno ▲▼	Upomínka (?) ▲▼	Příští upomínka		
000006	3m provazu	2007-04-27	A 326	-	-	Upravit	Smazat
000004	5kg TNT	2005-11-08	Honza Bajeru	12	2009-11-08	Upravit	Smazat
000008	Auto-svarečka	2008-12-08	Veselý Lukáš	-	-	Upravit	Smazat
000777	Brambory	2008-12-11	Novák Jan	6	2009-06-11	Upravit	Smazat
000007	Elektrická kytara	2002-05-07	Frainchmanová Rita	1	2009-05-07	Upravit	Smazat
546954	Hasicí přístroj	2008-12-20	A 1035	6	2009-06-20	Upravit	Smazat
000042	Kamna na olej	2008-12-08	NA 1	60	2013-12-08	Upravit	Smazat
000779	Klávsnice	2008-12-11	A 1023	1	2009-05-11	Upravit	Smazat
000011	Klávsnice Genius KB21e	2008-12-08	D 931	-	-	Upravit	Smazat
000002	Notebook ASUS S12	2008-12-03	D 931	6	2009-06-03	Upravit	Smazat
000001	Optická myš Trust	2008-12-04	A 326	-	-	Upravit	Smazat
000005	Vagon uhlí	2008-07-07	Veselý Lukáš	-	-	Upravit	Smazat
000780	Vit'a	2008-12-14	Honza Bajeru	1	2009-05-14	Upravit	Smazat
000010	Žehlící prkno	2008-12-08	D 931	60	2013-12-08	Upravit	Smazat

© 2009 Lukáš Veselý

Obrázek 16 - Výpis evidovaného majetku

- **Krok 3** – Uživatel se přepne do sekce “Čárové kódy”. Může si vybrat ze tří možností jejich generování. Pro příklad využije první možnost. První generovaný kód má číslo nejvyššího kódu zaevidovaného v databázi inkrementovaného o jedna.

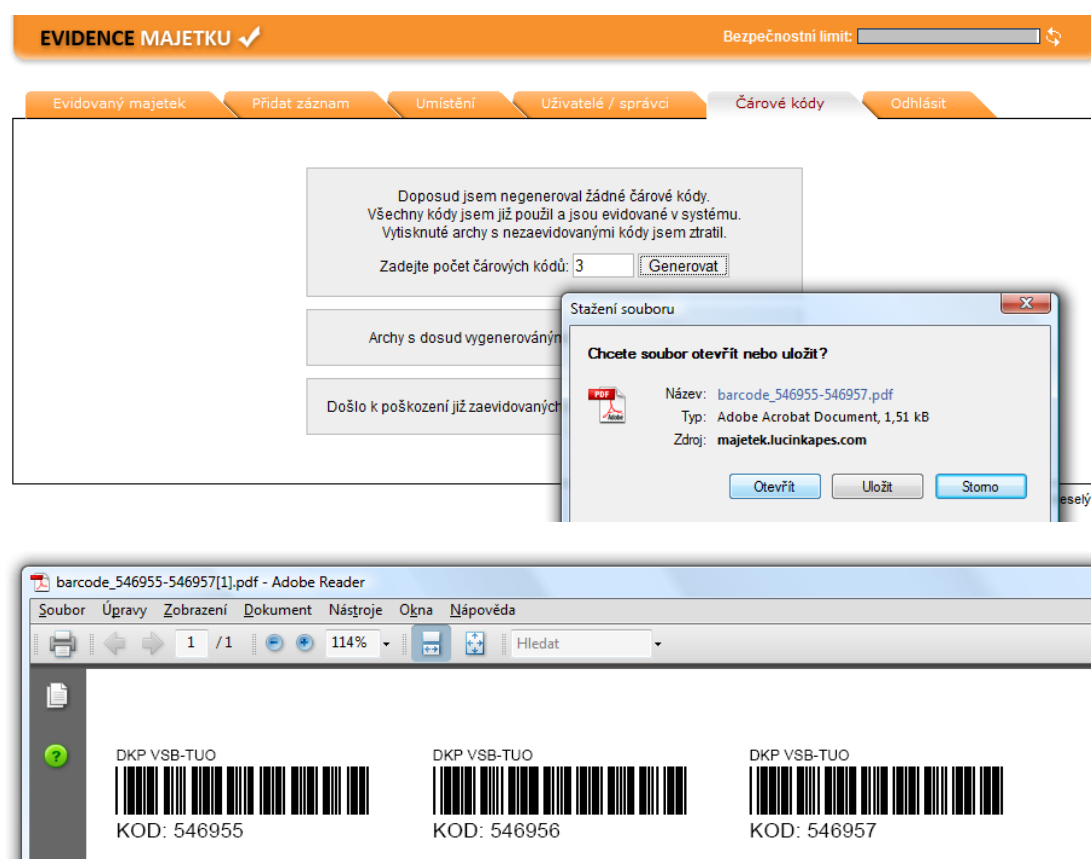
Druhá možnost generuje první kód od čísla nejvyššího kódu generovaného naposledy inkrementovaného o jedna. Toto číslo je uloženo v XML souboru `settings.xml` a po vygenerování nové sekvence se nahradí číslem novým, aktuálním.

Soubor: /settings.xml

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <settings>
3   <barcode>
4     <highest_printed>853</highest_printed>
5   </barcode>
6 </settings>
```

V poslední možnosti může uživatel nechat vygenerovat libovolný počet kódů, maximálně 52, tj. jedna strana A4. Jedině u této možnosti může ovlivnit jejich rozmístění na stránce.

Uživatel zadá počet kódů, které požaduje a stiskne tlačítko “Generovat”. Zobrazí se mu nabídka, zda chce PDF soubor s kódy přímo otevřít nebo jen uložit. Název souboru pro přehlednost obsahuje číslo nejnižšího a nejvyššího obsaženého kódu. Vybere si v tomto případě “Otevřít”.



Obrázek 17 - Požadavek na generování kódů s ukázkou výsledného PDF souboru

4.11. Verze pro mobilní zařízení

Jedním z požadavků na můj systém je i jeho přístupnost přes mobilní zařízení. Původně jsem plánoval udělat jednoduchou Java aplikaci, která by pracovala s MySQL databází a byla nainstalována například v mobilním telefonu. Narazil jsem zde na dva problémy.

Prvním je relativní neuniverzálnost řešení. Relativní píši proto, že většina mobilních telefonů Javou sice disponuje, ale na rozdíl od internetového prohlížeče není v ostatních zařízeních tak rozšířená.

Druhým problémem je umístění celého testovacího systému na placeném webhostingu. Standardně nejsou databáze na jejich serverech přístupny zvenku. Technická podpora na žádost sice povoluje tento přístup, ale jen z jedné pevné veřejné IP adresy. S požadavky na provoz mého systému je to neslučitelné.

Vytvořil jsem proto jednoduché webové rozhraní. Funguje na stejných principech jako rozhraní pro osobní počítače, má ale omezenou funkčnost a prezentační vrstvu.

Jaký je obecný rozdíl mezi klasickými stránkami a zmenšenou verzí pro mobilní telefony a jiná zařízení?

- stránky se na mobil načítají rychleji,
- struktura stránky je přizpůsobena displeji telefonu,
- zmenšené stránky obsahují funkce umožňující snazší orientaci,
- stránky jsou nazývané také jako wapové stránky.

Při tvorbě této verze stránek jsem použil testovací emulátor z webové stránky <http://www.mobilnaut.cz/?wap>.

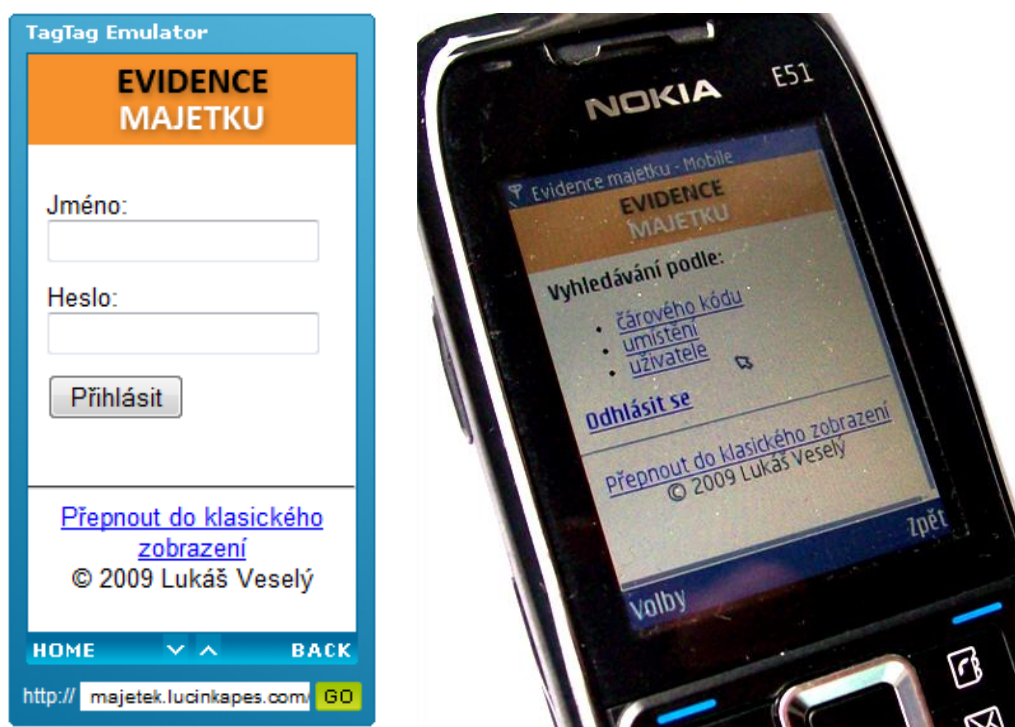
Funkce implementované ve verzi rozhraní pro mobilní zařízení jsou:

- Přihlášení do systému,
- Vyhledání majetku podle zadání čísla čárového kódu,
- Vyhledání majetku podle zadání budovy a místnosti, na kterou je evidován,
- Vyhledání majetku podle uživatele, na kterého je evidován.

O tom, zda bude načtena klasická verze nebo verze pro mobilní zařízení, rozhoduje JavaScript pracující s objektem `navigator`. Standardně se používá pro detekci prohlížeče uživatele, například pro vedení statistiky detailů o návštěvnících stránek. U objektu `navigator` jsem využil vlastnost `platform` a `userAgent`. V těchto řetězcích jsou vyhledávána klíčová slova, která označují mobilní zařízení nebo telefon (např: `nokia`, `symbian`, `mobile`, `alcatel`, `opera mini`, `playstation`, `psp`, `htc`, `tmobile`, `samsung`, `blackberry` a další). Je-li alespoň v jednom řetězci jedno z uvedených slov nalezeno, je zobrazena verze systému pro mobilní zařízení. Taková kontrola není vždy jednoznačná, a proto jsem využil ještě prostou kontrolu rozlišení obrazovky zařízení. Je-li rozlišení menší než 300 x 300px, je zobrazena tato verze také. Ani druhá kontrola nevylučuje určité procento možnosti vzniku chyby. Verze systému pro mobilní zařízení proto

obsahuje odkaz, umožňující přepnutí do klasické verze. Naopak uživatel klasické verze si může verzi pro mobilní zařízení zobrazit modifikací URL adresy (přidá do ní řetězec /mobile).

Na obrázku č. 17 je ukázka výsledného provedení při zobrazení přes wapový emulátor a mobilní telefon Nokia E51.



Obrázek 18 - Ukázka grafického rozhraní pro mobilní zařízení

5. Rozšiřitelnost systému

Primárním účelem mého systému byla implementace generování a tisku čárových kódů za pomoci PHP a jeho provedení jako online řešení pro počítače i mobilní zařízení. Do informačního systému je možné v budoucnu dodatečně doprogramovat mnoho dalších funkcí. Uvedu několik z nich:

- **Návaznost na účetnictví a platné zákony.** Systém je vytvořen se záměrem evidovat drobný majetek. Z hlediska účetnictví takový dlouhodobý hmotný majetek podniku, jehož cena je nižší než zákonem stanovená hranice 40.000,- Kč a doba použitelnosti delší než 1 rok. (Nehmotný a finanční majetek, nemůžeme označit čárovým kódem). Tento majetek je možné členit do odpisových skupin, klasifikovat ho podle tzv. “Klasifikace majetku“, podle zákona o daních z příjmu, evidovat jeho technické zhodnocení nebo implantovat podporu leasingu. To v případě, že se nejedná o majetek vlastní, ale pronajatý. S tím je spojená i případná evidence pronajímatelů.
- **Identifikace za pomoci bezkontaktních čidel.** O této technologii jsem se zmiňoval v jedné z předešlých kapitol. Jedná se o doplňkové označení vhodné zejména pro drobný majetek s vysokou hodnotou. Využití tohoto značení je dvojí. Kromě samotné identifikace je za pomoci dalších zařízení možné zabránit odcizení předmětu. RFID čipy bez toho, i když nenesou žádnou konkrétní informaci, se dnes již využívají v prodejnách textilu, drobné elektroniky nebo knih jako ochrana před zloději. Konkrétní místnosti nebo vchody do budov je nutné vybavit detekčními rámy s alarmem.
- **Evidence zapůjčení.** Není nutná implementaci veškerého kompletního výpůjčního systému jaký používají půjčovny či knihovny. Stačí u majetku označit, že byl dočasně přemístěn na jiné místo než je původně evidován, případně že byl zapůjčen jiné osobě, která není jeho uživatelem.
- **Uživatelská nastavení.** Současná podoba systému neumožňuje uživateli ovlivnit příliš výsledný tiskový výstup a pracuje jen s jedním typem samolepících štítků. Je vhodné implementovat více typů štítkových archů. Kódy by se tiskly jak v jiném rozmístění, tak i v jiné velikosti. Dále je možné uživateli umožnit, aby ovlivnil doplňující text kódu.

6. Závěr

PHP skriptovací jazyk existuje již velmi dlouhou dobu. I přes množství mnohem mladších a modernějších nástupců pro tvorbu dynamických webových aplikací se bude určitě používat ještě řadu let. Svědčí o tom také stále vznikající nové frameworky, které zjednodušují a sjednocují často řešené úlohy, např. frameworky Zend, Prado nebo Nette.

Stejná situace je i v oblasti čárových kódů. Nejde o žádnou novinku, ale systémy jejich čtení se stále zdokonalují. Vyvíjejí se nové typy kódů, a to zejména mezi tzv. 2D kódy. Jako příklad jsem uváděl kód QR, který se u nás začíná rozšiřovat. Existují i další, jako např. kruhový ShotCode, barevný MicrosoftCode nebo kód BeeTag, připomínající včelí plástev. Budoucnost tak nepatří výhradně RFID bezkontaktním čidlům. Podle mého názoru půjde o vzájemnou symbiózu a obě technologie se budou doplňovat.

Masovému rozšíření nových technologií brání dnes slabá standardizace. Netýká se to jen prostředků pro automatizovaný sběr dat, ale většiny odvětví z oblasti informačních technologií.

Výsledkem mé práce je systém, který je možné nasadit k reálnému použití v menší organizaci nebo firmě. Je však nutné provést drobné optimalizační úpravy podle konkrétních potřeb subjektu. Celé řešení je cenově dostupné a velmi rychle realizovatelné. Hlavní úkol, a to usnadnit evidenci majetku, byl splněn.

7. Použité zdroje

- [1] **Adams Communications**. *All About Code 39 Barcode* [online]. c1995 , 02/18/2009 [cit. 2010-01-05]. Dostupný z WWW: <<http://www.adams1.com/39code.html>>.
- [2] **BELOS Trade, s. r. o.**. *Čárový kód* [online]. c2004 [cit. 2010-04-18]. Dostupný z WWW: <<http://www.carovykod.com/index.php?id=2/cz>>.
- [3] **EPS, Neil Van**. *Drawing Barcodes in Windows Part 1 - Code 39* [online]. c2002 , 14 Jun 2002 [cit. 2010-01-15]. Dostupný z WWW: <<http://www.codeproject.com/KB/graphics/barcode1.aspx>>.
- [4] **GRAMES, Martin**. *Hackerem sám sobě (1) - PHP Injection* [online]. c2006-2008 , 26. 7. 2007 [cit. 2010-04-22]. Dostupný z WWW: <<http://www.chapadlo.cz/weblog/clanek/hackerem-sam-sobe-1-php-injection>>.
- [5] **GRIMMICH, Šimon**. *PHP /základy/* [online]. 2003-2008 [cit. 2010-04-20]. Dostupný z WWW: <<http://www.tvorba-webu.cz/php/>>.
- [6] **JAHODOVÁ, Radka**. *Dům budoucnosti – využití IT v domácnosti* [online]. 2003 , 30.12. 2003 [cit. 2010-04-22]. Dostupný z WWW: <<http://www.fi.muni.cz/usr/jkucera/pv109/2003p/xjahodov.htm>>.
- [7] **JANOVSKÝ, Dušan**. *Objekt window* [online]. [2002-2009] , 17. dubna 2009 [cit. 2010-05-03]. Dostupný z WWW: <<http://www.jakpsatweb.cz/javascript/objekt-window.html#navigator>>.
- [8] **KLEGA, Vratislav**. *QR kódy: Sejmout a na web* [online]. [2009] , 07.05.2008 [cit. 2010-04-25]. Dostupný z WWW: <<http://www.chip.cz/cs/clanky/qr-kody-sejmout-a-na-web.html>>.
- [9] **KOCMAN, Jiří**. *Jak na démona Cron* [online]. [2002] , 21. 4. 2002 [cit. 2010-04-18]. Dostupný z WWW: <<http://interval.cz/clanky/jak-na-demonu-cron/>>.
- [10] **KREJČÍ, Aleš**. *Kódy, čtečky a jejich využití* [online]. [2009] , 18. 3. 2009 [cit. 2010-04-25]. Dostupný z WWW: <<http://www.elaldio.eu/95-kody-ctecy-a-jejich-vyuziti>>.
- [11] **NICHOLLS, Stu**. *Overlapping tabs - Type 2* [online]. c2005 , 16th February 2007 [cit. 2010-01-15]. Dostupný z WWW: <<http://www.cssplay.co.uk/menus/overlap.html?current=one>>.
- [12] **ParCon & Partner, v. o. s.**. *Slovníček účetních pojmů* [online]. 2006-2009 , 28.06.2008 [cit. 2010-04-25]. Dostupný z WWW: <<http://www.testyzucetnictvi.cz/slovnicek-ucetnich-pojmu.php?pojem=dkp>>.
- [13] **RŮŽIČKA, Pavel**. *Začínáme používat sessions v PHP* [online]. [2002] , 21. 11. 2002 [cit. 2010-01-15]. Dostupný z WWW: <<http://interval.cz/clanky/zaciname-pouzivat-sessions-v-php/>>.
- [14] **Seznam.cz, a.s.**. *QR kódy* [online]. c1996-2009 [cit. 2010-04-25]. Dostupný z WWW: <<http://www.smobil.cz/qrcode/popis>>.

- [15] **SNÍŽEK, Martin.** *AJAX – kde jsou hranice?* [online]. c2005-2009 , 13.9.2005 [cit. 2010-04-20]. Dostupný z WWW: <<http://www.snizekweb.cz/clanky/ajax-kde-jsou-hranice/>>.
- [16] **Vodafone Czech Republic a.s.** . *Zmenšení stránek v mobilním telefonu* [online]. c2009[cit. 2010-04-05]. Dostupný z WWW: <http://www.vodafone.cz/osobni/data/data_optimalizace_mobil.htm>.
- [17] **VOJÁČEK, Petr.** *SQL Injection a zabezpečení* [online]. c2004-2009 , 23. 04. 2007 [cit. 2010-04-22]. Dostupný z WWW: <<http://programujte.com/index.php?akce=clanek&cl=2007041802-sql-injection-a-zabezpeceni>>.
- [18] **VRÁNA, Jakub.** *Obrana proti SQL Injection* [online]. c2005-2008 , 2.3.2005 [cit. 2010-04-22]. Dostupný z WWW: <<http://php.vrana.cz/obrana-proti-sql-injection.php>>.
- [19] **YANG, Xin.** *Chained Selects Step-by-step: Implement the list group* [online]. c2004 [cit. 2010-01-15]. Dostupný z WWW: <<http://www.yxscripts.com/cs/examples/listgroup.html>>.
- [20] **ZACHAR, Jiří.** *FPdf: jak na české znaky* [online]. c2009 [cit. 2010-01-15]. Dostupný z WWW: <<http://www.zaachi.com/cs/items/fpdf-jak-na-ceske-znaky.html>>.
- [21] *CCD Barcode Scanner-2.jpg* [online]. c2009 , 13. August 2007 [cit. 2010-04-22]. Dostupný z WWW: <http://commons.wikimedia.org/wiki/File:CCD_Barcode_Scanner-2.jpg>.
- [22] *Flickr - Photo Sharing!* [online]. c2009 , 2008:05:26 [cit. 2010-01-15]. Dostupný z WWW: <<http://www.flickr.com/photos/larskflem/2527259341/>>.
- [23] *MySQL manuál* [online]. c2003-2005 [cit. 2010-01-05]. Dostupný z WWW: <<http://mm.gene.cz/>>.
- [24] *Radio Frequency Identification* [online]. c2008 , January 13, 2008 [cit. 2010-04-25]. Dostupný z WWW: <http://dcm-abs.blogspot.com/2008_01_01_archive.html>.
- [25] *FPDF Library PDF Generator* [online]. [2002-2009] [cit. 2010-01-15]. Dostupný z WWW: <<http://fpdf.org/>>.
- [26] *User-Agent a rozpoznání prohlížeče* [online]. [2008] , 31.10.2008 [cit. 2010-04-22]. Dostupný z WWW: <<http://www.mobilniweb.info/user-agent-rozpoznani-prohlizece/57>>.
- [27] **DOČEKAL, Daniel.** *Budoucnost a současnost „čarových“ kódů pro mobily* [online]. c2008 [cit. 2010-04-20]. Dostupný z WWW: < <http://www.lupa.cz/clanky/budoucnost-a-soucasnost-carovych-kodu-pro-mobily/>>.

8. Přílohy

8.1. Příloha č. 1 – Datový slovník

Tabulka majetek

Entita	Datový typ	Klíč	Null	Výchozí	Index	Poznámka a IO
barcode	integer	PK	-	-	A	Čárový kód
nazev	varchar(100)	-	-	-	A	Název evidovaného předmětu
pridano	date	-	-	-	-	Datum přidání (RRRR-MM-DD)
uzivatel	integer	FK	A	-	-	Id držitele předmětu
umisteni	integer	FK	A	-	-	Id místnosti, kde se nachází
perioda_upominky	integer	-	A	0	-	Perioda upomínky v měsících
vyrazeno	datetime	-	A	NULL	-	Není využíváno

Tabulka majetek_vyrazeny

Entita	Datový typ	Klíč	Null	Výchozí	Index	Poznámka a IO
barcode	integer	PK	-	-	A	Čárový kód
nazev	varchar(100)	-	-	-	A	Název evidovaného předmětu
pridano	date	-	-	-	-	Datum přidání (RRRR-MM-DD)
uzivatel	integer	FK	A	-	-	Id držitele předmětu
umisteni	integer	FK	A	-	-	Id místnosti, kde se nachází
perioda_upominky	integer	-	A	0	-	Perioda upomínky v měsících
vyrazeno	datetime	-	-	-	-	Datum vyřazení majetku

Tabulka majetek_editace

Entita	Datový typ	Klíč	Null	Výchozí	Index	Poznámka a IO
id	integer	PK	-	-	A	Id záznamu
barcode	integer	FK	-	-	A	Čárový kód edit. majetku
nazev	varchar(100)	-	-	-	A	Název evidovaného předmětu
pridano	date	-	-	-	-	Datum přidání (RRRR-MM-DD)
uzivatel	integer	FK	A	-	-	Id držitele předmětu
umisteni	integer	FK	A	-	-	Id místnosti, kde se nachází
perioda_upominky	integer	-	A	0	-	Perioda upomínky v měsících
vyrazeno	datetime	-	A	NULL	-	Není využíváno

Tabulka majetek_log

Entita	Datový typ	Klíč	Null	Výchozí	Index	Poznámka a IO
id	integer	PK	-	-	A	Id záznamu
uzivatel_id	integer	FK	-	-	A	Id uživatele
ip_adresa	varchar(20)	-	-	-	-	Ip adresa přihlašeného uživatele
datum_cas	datetime	-	-	-	-	Datum přihlášení

Tabulka majetek_uzivatele

Entita	Datový typ	Klíč	Null	Výchozí	Index	Poznámka a IO
id	integer	PK	-	-	-	Id
login	varchar(15)	-	-	-	A	Uživatelské jméno
heslo	varchar(32)	-	-	-	-	Heslo šifrované MD5
jmeno	varchar(25)	-	-	-	A	Křestní jméno
prijmeni	varchar(25)	-	-	-	A	Příjmení
email	varchar(50)	-	-	-	-	E-mailová adresa
opraveni	integer	-	-	0	-	0 = běžný uživatel, 1= admin
posledni_prihlaseni	datetime	-	A	NULL	-	Datum posledního přihlášení
pocet_prihlaseni	Integer	-	A	0	-	Počet přihlášení do systému

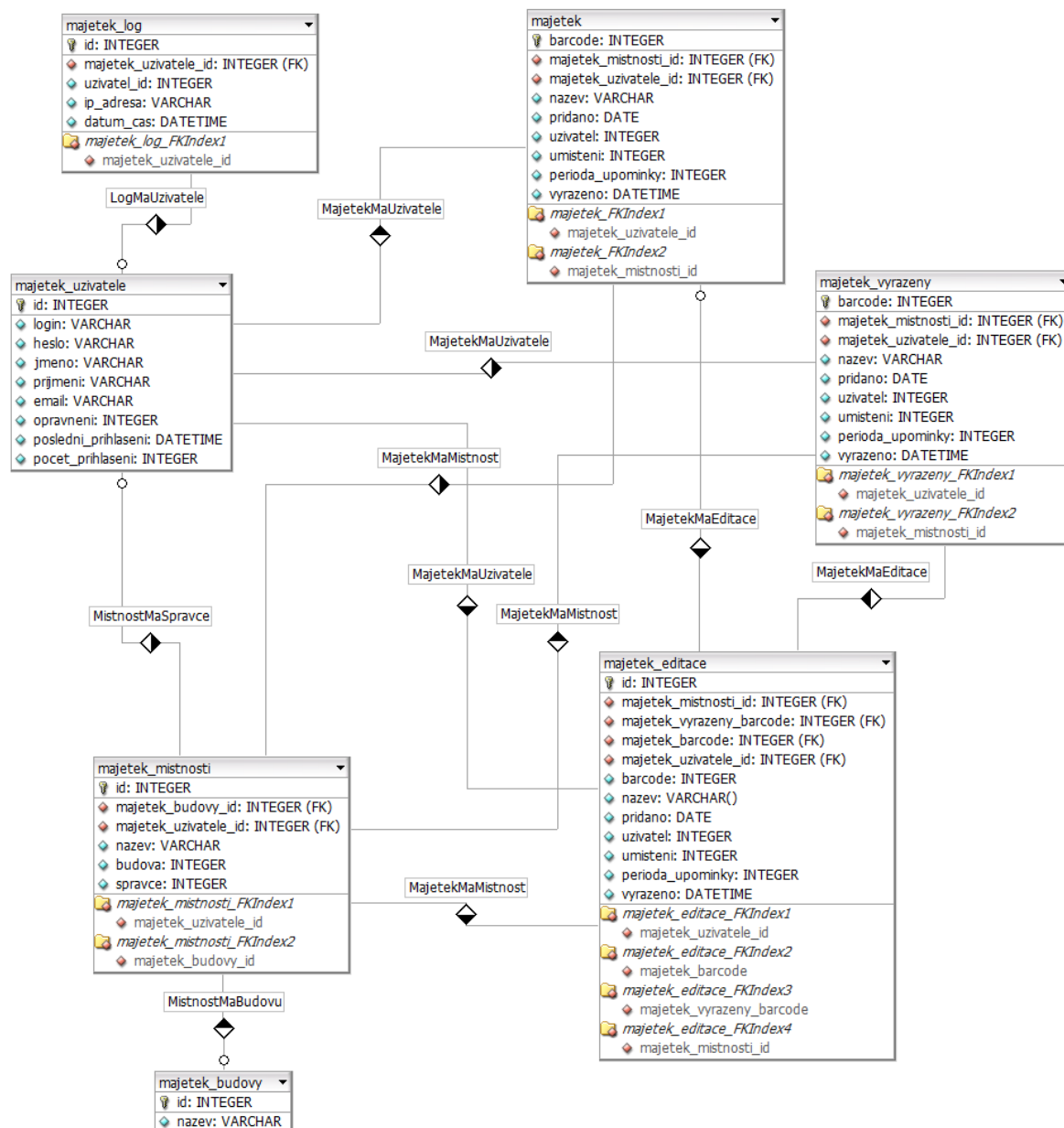
Tabulka majetek_budovy

Entita	Datový typ	Klíč	Null	Výchozí	Index	Poznámka a IO
id	integer	PK	-	-	A	Id
nazev	varchar(30)	-	-	-	A	Název budovy

Tabulka majetek_mistnosti

Entita	Datový typ	Klíč	Null	Výchozí	Index	Poznámka a IO
Id	integer	PK	-	-	A	Id
nazev	varchar(25)	-	-	-	-	Název místnosti
budova	integer	FK	-	-	-	Id budovy, kde se nachází
spravce	integer	FK	-	-	-	Id správce místnosti

8.2. Příloha č. 2 – ER diagram



Legenda:

- ◊ Kardinalita 1:N
- ◈ Kardinalita N:1
- ◊ Kardinalita 1:1
- Povinnost členství: ANO
- Povinnost členství: NE

Vytvořeno v DB Designer 4